

## EFFICIENT VALIDATION OF NETWORK CONFIGURATION CHANGE COMMANDS

### FIELD OF THE INVENTION

[0001] The present invention relates generally to storage management in a distributed network environment and, more particularly, to an efficient method for validating network change commands in a storage area network.

### BACKGROUND OF THE INVENTION

[0002] Management of storage resources is an essential aspect of implementing and maintaining storage area networks. Storage management may encompass a broad range of storage functions including file access, volume management and data backup. Of particular interest, storage management includes the ability to dynamically allocate individual storage units to shared storage groups and/or specific storage servers.

[0003] Dynamically allocating storage units is typically achieved through change commands issued from one or more of the client workstations having access to the storage area network. Because change commands can be scheduled for execution at a later time and can be issued (possibly concurrently) by several client workstations, it is important to verify the appropriateness of each change command before the request is implemented in the network. Specifically, it is necessary to determine if the desired changes embodied in the change commands are expressed in terms of the current network configuration.

**[0004]** Therefore, it is desirable to provide an improved method for validating network change commands in a storage area network.

#### SUMMARY OF THE INVENTION

**[0005]** In accordance with the present invention, an improved method is provided for validating network configuration change commands in a storage area network. The improved method includes: providing a change command to a network management device residing in the network, where the change command expresses a change to the configuration of the network and implicitly indicates an initial configuration state of the network therein; determining a current configuration state for the network; and comparing the initial configuration state indicated by the change command to the current configuration state of the network. The change command is implemented by the network management device when the initial configuration state indicated by the change command correlates to the current configuration state of the network.

**[0006]** For a more complete understanding of the invention, its objects and advantages, reference may be had to the following specification and to the accompanying drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0007]** Figure 1 is a block diagram of an exemplary distributed network environment that includes a storage area network in accordance with the present invention;

**[0008]** Figure 2 is a flowchart of an improved method for validating network change commands in accordance with the present invention;

**[0009]** Figure 3 is a flowchart of an exemplary validation algorithm for simple assign commands in accordance with the present invention; and

**[0010]** Figure 4 is a flowchart of an exemplary validation algorithm for complex assign commands in accordance with the present invention.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

**[0011]** Figure 1 illustrates a distributed network environment 10 that includes a storage area network 12. Specifically, the storage area network 12 includes one or more storage servers 14 as well as one or more storage devices 16. Each storage device 16 further includes a plurality of storage units. Storage servers 14 provide access to the storage units embodied in the storage devices 16 as is well known in the art. It is readily understood that the storage area network 10 may employ various network topologies. While the following description is provided with reference to a storage area network, it is readily understood that the present invention is applicable to other types of distributed data storage networks.

**[0012]** Management of storage resources is administered by a network management server 20. Change commands are typically used to express one or more requests to change the configuration of the network as is

well known in the art. Change commands may be issued from any one of various client workstations 22 residing in the network environment 10. The changes to the network configuration are then implemented by the network management server 20. The network management server 20 may be interconnected via a local area network (LAN) to the client workstations 22 and the storage servers 14. Because change commands can be scheduled for execution at a later time and can be issued by several client workstations (possibly concurrently), it is important to verify the appropriateness of each change command set. In particular, it is necessary to determine if the desired changes embodied in a change command are expressed in terms of the current network configuration.

[0013] In accordance with the present invention, an improved method is provided for validating change commands in a distributed network environment. The improved method relies upon the realization that any delta, such as a change command set, implicitly indicates an initial state. In other words, a change cannot exist autonomously, but rather exists relative to an initial state. Likewise, network change commands implicitly indicate an initial configuration state of the network. Thus, it is possible to check if the initial configuration state expressed in a given change command corresponds to the current configuration state of the network.

[0014] Referring to Figure 2, a change command set 32 is compared at step 34 to a virtual representation 36 of the physical network configuration. The virtual representation 36 is a detailed model of the physical state of the network as is well known in the art. Thus, the virtual representation 36 expresses the current configuration state of the network. If

the change command was defined relative to the current state of the network, then the change commands are implemented at step 38 by the network management server. On the other hand, if the change command does not correlate to the current state of the network, then the change command is deemed an invalid request. In this way, an improved method is provided for validating network change commands in a distributed network environment.

**[0015]** Storage allocation commands may be used to dynamically configure storage resources within a network environment. Storage allocation is generally implemented using two fundamental change commands: assign and unassign. When an individual storage unit is assigned to a storage server (also referred to as a host), the server is granted read-write access to the assigned storage unit. Conversely, when an storage unit is unassigned from a server, the storage unit is no longer available to the server.

**[0016]** Storage allocation is also facilitated through the use of security groups. Security groups enable the grouping of and access to related storage units. For illustration purposes, two different types of security groups are described below. However, it is envisioned that other types of groups are also within the broader aspects of the present invention.

**[0017]** First, shared groups allow multiple hosts to share the same set of storage units with read-write access. When a host is assigned to a shared group, all of the storage in the shared group is assigned to the host. When a host is unassigned from a shared group, all of the storage in the shared group is unassigned from the host. Similarly, when a storage unit is assigned to a shared group, it is automatically assigned to each host in the

shared group. When a storage unit is unassigned from a shared group, it is automatically unassigned from each host in the shared group.

[0018] Second, associated LUN groups are used to group storage units (also referred to as logical units) that can be assigned and unassigned as a group. For instance, associated LUN groups may be used to group stripe sets, mirror sets, and sets of storage units that contain parts of the same database. When an associated LUN group is assigned to a host or shared group, all of the storage units in the associated LUN group are assigned to the particular host or all of the hosts in the shared group. When an associated LUN group is unassigned from a host or shared group, all of the logical units in the associated LUN group are unassigned from the affected hosts. Similarly, if a storage unit is added to an associated LUN group that is assigned to a host or shared group, then the newly assigned storage unit is automatically assigned to the particular host or all of the hosts in the shared group. If the storage unit is unassigned from an associated LUN group, then the unassigned storage unit is automatically unassigned from the affected hosts.

[0019] To further illustrate the validation method of the present invention, a preferred implementation of the methodology as applied to these exemplary storage allocation commands is set forth below. Specifically, a validation method is provided for the following six assign commands: (1) assign storage unit to host; (2) assign storage unit to shared group; (3) assign storage unit to associated LUN group; (4) assign associated LUN group to host; (5) assign associated LUN group to shared group; and (6) assign host to shared group. To simplify the discussion, the first three assign commands are

referred to as “simple assign commands”; whereas the remaining three assign commands are referred to as “complex assign commands”.

**[0020]** Figure 3 illustrates a preferred validation algorithm for each of the simple assign commands. In each of these commands, an individual storage unit is being assigned to either a host, a shared group or an associated LUN group. Implicitly expressed in each of these commands is that the individual storage unit is not assigned or grouped in the current network configuration. Accordingly, the validation algorithm will confirm this configuration state for the individual storage unit.

**[0021]** First, determine the current configuration state for the storage unit at step 42. If the storage unit is currently assigned, then it is not available to be assigned (without first being unassigned) and thus the assign command is an invalid request. Conversely, if the storage unit is currently unassigned, then determine if the storage unit is grouped at step 44. If the storage unit is currently grouped, then it is not available to be assigned (independently from its grouping) and thus the assign command is invalid. However, if the storage unit is not currently assigned or grouped, then it may be assumed that the assign command is expressed in terms of the current network configuration, and therefore is a valid change command.

**[0022]** An additional validation algorithm for the complex assign commands is shown in Figure 4. Each of these commands involves the assignment of multiple storage units. Implicitly expressed in each of these commands is that each individual storage unit is part of either a shared group or an associated LUN group in the current network configuration.

Accordingly, the validation algorithm will confirm this configuration state for each individual storage unit.

**[0023]** Initially, each storage unit in a particular grouping is identified at step 52. For instance, in the case of an associated LUN group being assigned to a host, each storage unit in the LUN group is identified. In the case of an associated LUN group being assigned to a shared group, each storage unit in the LUN group is identified. Additionally, each host associated with the shared group is also identified. Lastly, in the case of the host being assigned to a shared group, each storage unit in the shared group is identified. The composition of an associated LUN group or a shared group is easily ascertained from the virtual representation of the physical network as is known in the art.

**[0024]** Next, the applicable storage unit – host combinations are iterated for each assign command at step 54. Each storage unit can then be checked to determine if it belongs to a group at step 56. If the storage unit is not currently grouped, then there has been a change in the network configuration and thus the assign command is invalid. If the storage unit is currently grouped, then the assign command is valid as it relates to this particular storage unit. This process is repeated for each storage unit identified for a particular grouping. If each of the storage units are currently grouped, then it may be assumed that the assign command is expressed in terms of the current network configuration, and therefore is a valid change command. One skilled in the art will readily recognize that other verification steps may also be incorporated into this validation algorithm.



[0025] Lastly, a validation algorithm may be implemented for an unassign command. An unassign command implicitly indicates that the storage unit to be unassigned is currently assigned within the network. Therefore, an exemplary validation technique is to determine if the storage unit is currently assigned within the network. If the storage unit is currently assigned, then it may be assumed that the assign command is expressed in terms of the current network configuration, and thus is a valid change command. On the other hand, if the storage unit is not currently assigned, then the command is invalid.

[0026] Although these various validation algorithms are presently preferred, they are not intended as a limitation on the broader aspects of the present invention. For instance, it is envisioned that other validation algorithms may be suitable for validating these particular storage allocation commands. Moreover, it is envisioned that other validation techniques based on change commands that are expressed in terms of the current network configuration are also within the scope of the present invention. In other words, while the invention has been described in its presently preferred form, it will be understood that the invention is capable of modification without departing from the spirit of the invention as set forth in the appended claims.